Easy introduction to MVC

A lecture by Burciu Iulian Gabriel

What is MVC?

MVC (Model-View-Controller) is a design pattern used in software development to **separate** an application into three main components: **Model**, **View** and **Controller**. It helps in **organizing code** by separating data handling, user interface and user interaction into distinct roles.

Model

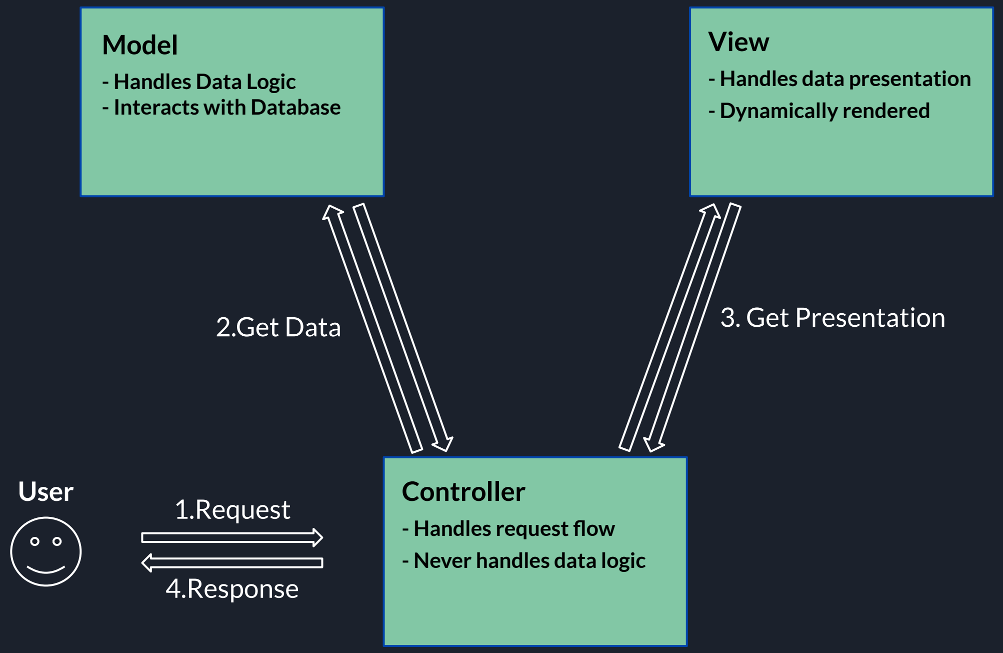
Think of the model as the “brain” of the application. It represents the data and the core logic of your program. It’s responsible for things like handling data, databases and ensuring that your application works correctly.

View

The “View” is responsible for the user interface and how information is presented to the user. The view is like the “face” of your application. It’s in charge of displaying information dynamically based on the data the Controller sends.

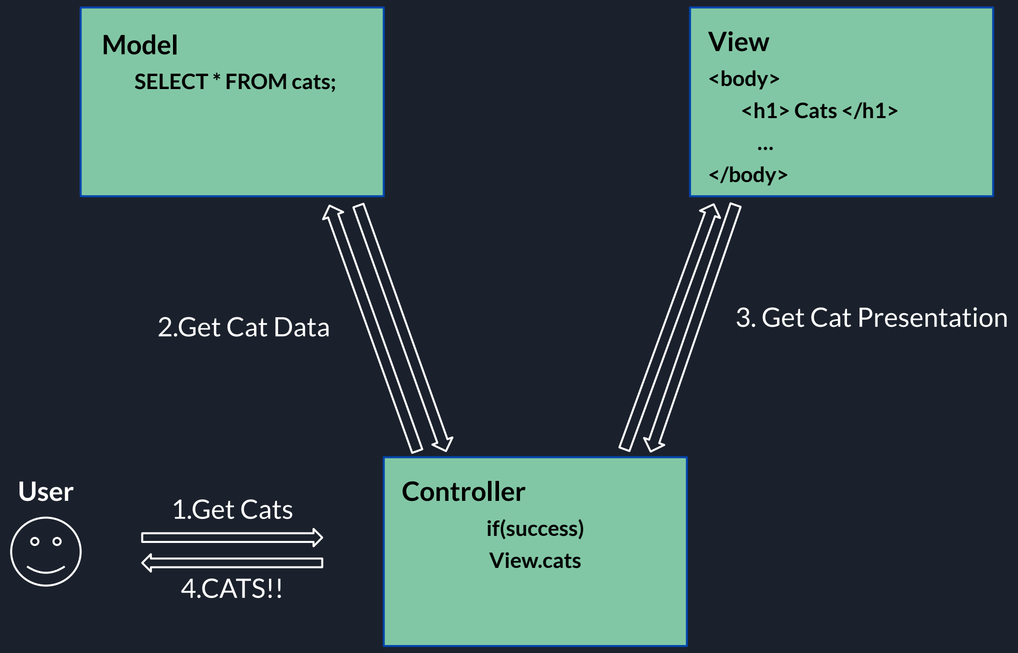
Controller

The Controller is the middleman between the Model and the View (The Model and the View never interact with each other). It takes user input from the View, processes it, communicates with the Model to fetch or update data, and then updates the View to reflect any changes. Essentially, it controls how the Model and the View interact.



Example taken from
Web Dev Simplified

MVC Example



1. User sends a request to a server to get a list of cats. The server will send that request to the Controller that handles cats.

2. The Controller will ask the Model that handles cats to return a list of all cats. The model will then query the database for a list of all cats and then return it to the Controller.

3. If the response from the Model is successful then the Controller will ask the View associated with cats to return a presentation of the list of cats. The view will take the list from the Controller and render the list into HTML that could be used by the browser.

4. The Controller will then take the presentation and return it back to the user, ending the request.

Why should I use MVC?

If you find yourself asking this question, here are some reasons why you should use MVC:

1. Separation of Concerns: MVC encourages a clear separation of responsibilities, making it easier to understand, modify and maintain the codebase.

2. Reusability: MVC promotes the reuse of code components. For example, you can use the same Model with different Views and Controllers, reducing Development time and effort.

3. Maintainability: With MVC, debugging and maintaining the application becomes more straightforward. Changes to one component are less likely to impact the others, resulting in more stable and dependable software.

4. Testing: The separation of components in MVC simplifies the testing process. Each component can be tested independently, resulting in more reliable and error-free code.

5. Scalability: MVC supports the growth and scalability of applications. It allows for the addition of new features or the expansion of the application without extensive changes to existing code, saving time and resources.

Thank you!

Contact:

IulianGabrielBIG@gmail.com

https://github.com/IulianGabriel